We claim:

1. A method for establishing a bound on the execution time of an application due to task interference in an instruction cache shared by a plurality of tasks, said method comprising the steps of:

determining a number of live frames of said application that are coexistent during execution of said application; and

establishing said bound based on said number of live frames.

2. The method of claim 1, wherein said number of live frames is a number of cache frames that contain a block that is accessed by said application in the future without an intervening eviction.

3. The method of claim 1, wherein said number of live frames is determined by a post-execution analysis of cache access patterns of said application.

4. The method of claim 1, wherein said number of live frames is determined by a run-time analysis of cache access patterns of a simulation of said application.

5. The method of claim 1, wherein said step of establishing said bound further comprises the step of comparing the sets that contain live frames of said application with sets accessed by an interrupting task to determine a maximum number of live-frames that may be affected by an interrupting task.

6. The method of claim 1, wherein said step of establishing said bound further comprises the steps of determining an effect of an interrupt at each possible interrupt point and establishing said bound based on a maximum of said effect of an interrupt at each possible interrupt point.

7. A system for establishing a bound on the execution time of an application due to task interference in an instruction cache shared by a plurality of tasks, said system comprising:

a memory that stores computer-readable code; and

a processor operatively coupled to said memory, said processor configured to implement said computer-readable code, said computer-readable code configured to:

determine a number of live frames of said application that are coexistent during execution of said application; and

establish said bound based on said number of live frames.

8. The system of claim 7, wherein said number of live frames is a number of cache frames that contain a block that is accessed by said application in the future without an intervening eviction.

9. The system of claim 7, wherein said number of live frames is determined by a post-execution analysis of cache access patterns of said application.

10. The system of claim 7, wherein said number of live frames is determined by a run-time analysis of cache access patterns of a simulation of said application.

11. The system of claim 7, wherein said processor is further configured to compare the sets that contain live frames of said application with sets accessed by an interrupting task to determine a maximum number of live-frames that may be affected by an interrupting task.

12. The system of claim 7, wherein said processor is further configured to determine an effect of an interrupt at each possible interrupt point and establish said bound based on a maximum of said effect of an interrupt at each possible interrupt point.

13. An article of manufacture for establishing a bound on the execution time of an application due to task interference in an instruction cache shared by a plurality of tasks, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to determine a number of live frames of said application that are coexistent during execution of said application; and

a step to establish said bound based on said number of live frames.

14. The article of manufacture of claim 13, wherein said computer readable program code means further comprises a step to compare the sets that contain live frames of said application with sets accessed by an interrupting task to determine a maximum number of live-frames that may be affected by an interrupting task.

15. The article of manufacture of claim 13, wherein said computer readable program code means further comprises a step to determine an effect of an interrupt at each possible interrupt point and establish said bound based on a maximum of said effect of an interrupt at each possible interrupt point.

16. A system for establishing a bound on an effect of task interference on an application in an instruction cache shared by a plurality of tasks, said system comprising:

means for determining a number of live frames of said application that are coexistent during execution of said application; and

means for establishing said bound based on said number of live frames.

17. The system of claim 16, wherein said number of live frames is determined by a post-execution analysis of cache access patterns of said application.

18. The system of claim 16, wherein said number of live frames is determined by a run-time analysis of cache access patterns of a simulation of said application.

19. The system of claim 16, further comprising means for comparing the sets that contain live frames of said application with sets accessed by an interrupting task to determine a maximum number of live-frames that may be affected by an interrupting task.

20. The system of claim 16, further comprising means for determining an effect of an interrupt at each possible interrupt point and establish said bound based on a maximum of said effect of an interrupt at each possible interrupt point.